

# CreateThread

Use restrictive permissions when creating new threads

Sean Barnum, Digital, Inc. [vita<sup>1</sup>]

Copyright © 2007 Digital, Inc.

2007-03-20

## Part "Original Digital Coding Rule in XML"

Mime-type: text/xml, size: 6821 bytes

<b>Attack Category</b>	<ul style="list-style-type: none"><li>• Privilege Exploitation</li></ul>				
<b>Vulnerability Category</b>	<ul style="list-style-type: none"><li>• Process management</li><li>• Unconditional</li></ul>				
<b>Software Context</b>	<ul style="list-style-type: none"><li>• Threads and Processes</li></ul>				
<b>Location</b>					
<b>Description</b>	<p>When creating a new thread it is important not to set permissions that will allow untrusted code to write to the thread.</p> <p>The CreateThread() function creates a new thread. The lpThreadAttributes argument allows one to set the security attributes of the created thread. Threads with loose access rules are a serious security problem. Do not set permissions that could allow untrusted code to gain access.</p>				
<b>APIs</b>	<b>FunctionName</b>	<b>Comments</b>			
	CreateThread				
<b>MethodofAttack</b>	If an attacker can get a handle to your thread with some sort of write access, he can call SetThreadContext() to rewrite the thread context in order to affect some sort of stack smashing attack.				
<b>ExceptionCriteria</b>					
<b>Solutions</b>	<b>Solution Applicability</b>	<b>Solution Description</b>	<b>Solution Efficacy</b>		
	When creating a new thread.	Don't allow an attacker to rewrite the context of some thread you create. Protect thread with proper ACLs.	Effective. Requires understanding security model.		
<b>Signature Details</b>	HANDLE CreateThread( LPSECURITY_ATTRIBUTES lpThreadAttributes,				

1. [http://buildsecurityin.us-cert.gov/bsi/about\\_us/authors/35-BSI.html](http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html) (Barnum, Sean)

```

SIZE_T dwStackSize,
LPTHREAD_START_ROUTINE lpStartAddress,
LPVOID lpParameter,
DWORD dwCreationFlags,
LPDWORD lpThreadId
);

```

### Examples of Incorrect Code

```

SECURITY_ATTRIBUTES sa;
sa.bInheritHandle=TRUE;
sa.lpSecurityDescriptor=NULL;
/* set up a NULL __DACL__ (not
the same as a NULL Security
Descriptor, which
actually will take a reasonable
default) in our security
descriptor to allow anyone to
connect to the pipe server */
sa.lpSecurityDescriptor =
(PSECURITY_DESCRIPTOR)
malloc(SECURITY_DESCRIPTOR_MIN_LENGTH);
InitializeSecurityDescriptor(sa.lpSecurityDescriptor,
SECURITY_DESCRIPTOR_REVISION);
SetSecurityDescriptorDacl(sa.lpSecurityDescriptor,
TRUE, (PACL) NULL,
FALSE);
sa.nLength = sizeof(sa);
sa.bInheritHandle = TRUE;
sa.nLength=sizeof(SECURITY_ATTRIBUTES);

// Create MAX_THREADS worker
threads.
for( i=0; i<MAX_THREADS; i++ )
{
// Allocate memory for thread
data.
pData =
HeapAlloc(GetProcessHeap(),
HEAP_ZERO_MEMORY,
sizeof(MYDATA));
if( pData == NULL )
ExitProcess(2);
// Generate unique data for each
thread.
pData->val1 = i;
pData->val2 = i+100;
hThread[i] = CreateThread(
sa, // *** use specific security
attributes
0, // use default stack size
ThreadProc, // thread function
pData, // argument to thread
function
0, // use default creation flags

```

```

    &dwThreadId[i]); // returns the
    thread identifier

    // Check the return value for
    success.
    if (hThread[i] == NULL)
        ExitProcess(i);
    }
    // Wait until all threads have
    terminated.
    WaitForMultipleObjects(MAX_THREADS,
        hThread, TRUE, INFINITE);

    // Close all thread handles upon
    completion.
    for(i=0; i<MAX_THREADS; i++)
    {
        CloseHandle(hThread[i]);
    }
}

```

### Examples of Corrected Code

```

/* Specifying a NULL
lpThreadAttributes to
CreateThread() will apply a
default security descriptor to
the new thread. Normally, this
should be reasonably restrictive
and hence reasonably secure. But
check to confirm specifics. */

// Create MAX_THREADS worker
threads.
for( i=0; i<MAX_THREADS; i++ )
{
    // Allocate memory for thread
    data.
    pData =
        HeapAlloc(GetProcessHeap(),
            HEAP_ZERO_MEMORY,
            sizeof(MYDATA));
    if( pData == NULL )
        ExitProcess(2);
    // Generate unique data for each
    thread.
    pData->val1 = i;
    pData->val2 = i+100;
    hThread[i] = CreateThread(
        NULL, // default security
        attributes
        0, // use default stack size
        ThreadProc, // thread function
        pData, // argument to thread
        function
        0, // use default creation flags
        &dwThreadId[i]); // returns the
        thread identifier
}

```

```

    // Check the return value for
    // success.
    if (hThread[i] == NULL)
        ExitProcess(i);
    }
    // Wait until all threads have
    // terminated.
    WaitForMultipleObjects(MAX_THREADS,
    hThread, TRUE, INFINITE);

    // Close all thread handles upon
    // completion.
    for(i=0; i<MAX_THREADS; i++)
    {
        CloseHandle(hThread[i]);
    }
}

```

## Source References

- [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/processes\\_and\\_threads.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/processes_and_threads.asp)<sup>2</sup>
- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/Security/createthread>

## Recommended Resource

### Discriminant Set

<b>Operating System</b>	• Windows
<b>Languages</b>	• C • C++

## Cigital, Inc. Copyright

Copyright © Digital, Inc. 2005-2007. Digital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Digital, including information about “Fair Use,” contact Digital at [copyright@digital.com](mailto:copyright@digital.com)<sup>1</sup>.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

---

1. <mailto:copyright@digital.com>